Housekeeping

- Assignment 1 due on Tuesday May 23
- Project abstracts due today
Review

- Distance-based Classification
  - Use training samples to find defining features of known classes
  - Determine a suitable prototype for each class
  - Determine a suitable distance measure (metric)
  - Use the prototype(s) and metric to classify new unknown patterns to one of the classes

Minimum Euclidean Distance (MED)

\[(x - m_1)^T (x - m_1) \leq (x - m_2)^T (x - m_2)\]

\[
\begin{align*}
x &= \begin{bmatrix} 2 \\ 3 \end{bmatrix} & m_1 &= \begin{bmatrix} 1 \\ 2 \end{bmatrix} & m_2 &= \begin{bmatrix} 5 \\ 3 \end{bmatrix} \\
& & & & C_1 \leq C_2 & & & & C_1 \leq C_2 \\
\end{align*}
\]

Classify \(x\) to \(C_1\)
Review

Minimum Intra-Class Distance (MICD)
A.K.A Mahalanobis Distance

\[
(x - m_1)^T S_1^{-1} (x - m_1) \leq C_1
\]

\[
(x - m_2)^T S_2^{-1} (x - m_2) \leq C_2
\]

\[
\begin{bmatrix}
2 \\
3
\end{bmatrix}
\begin{bmatrix}
1 & 1 \\
1 & 2
\end{bmatrix}
\begin{bmatrix}
2 & -3 & 3 \\
-3 & 5 & 1
\end{bmatrix}
\begin{bmatrix}
1 \\
1
\end{bmatrix}
\begin{bmatrix}
C_1 \\
C_2
\end{bmatrix}
\leq
\begin{bmatrix}
-4 & -1 \\
-1 & 1
\end{bmatrix}
\begin{bmatrix}
0.5 & 0.2 \\
0 & 0.2
\end{bmatrix}
\begin{bmatrix}
-4 \\
-1
\end{bmatrix}
\]

\[
\begin{bmatrix}
5 & 3 \\
3 & 2
\end{bmatrix}
\begin{bmatrix}
2 \\
0
\end{bmatrix}
\]

\[
\begin{bmatrix}
2 & 0 \\
0 & 5
\end{bmatrix}
\]

Classify \( x \) to \( C_1 \)

Review

Unit Standard Deviation Contour
A way to visualize the distribution

Calculate the eigenvectors and eigenvalues from the covariance matrix (S)

![Figure 2.9 From Pattern Classification by Duda, Hart, and Stork](image-url)
Review

Decision surface for 2 classes in multidimensional feature space

\[(x - m_1)^T S_1^{-1} (x - m_1) - (x - m_2)^T S_2^{-1} (x - m_2) = 0\]

\[\bar{x}^T Q \bar{x} + w^T \bar{x} + w_0 = 0\]

HyperQuadratic

If \(S_1 = S_2\), then \(Q=0\) and decision surface becomes a hyperplane:

\[(m_2 - m_1)^T S^{-1} \left( \frac{x - (m_1 + m_2)}{2} \right) = 0\]

Bayesian Classification
Recall the problem with the MICD

The MICD always favours $C_2$ over $C_1$, even close to their common mean, because the MICD scales the distributions so that they are the same.

Ideally, we want to favour the class with the highest probability for the given pattern:

$$P(C_i | \mathbf{x}) \frac{C_i}{C_j} \geq P(C_j | \mathbf{x})$$

where $P(C_i | \mathbf{x})$ is the ‘a posterior’ (after measurement) probability of class $C_i$ given $\mathbf{x}$. 
Maximum A Posterior (MAP) Classifier

To get \( P(C_i|x) \), we need \( p(x|C_i) \) and Bayes’ Theorem:

\[
P(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)P(C_i)}{p(\mathbf{x})}
\]

Where \( p(x|C_i) \) is the class conditional probability density (p.d.f.), which needs to be estimated from the available samples or otherwise assumed.

\( P(C_i) \) is the ‘a priori’ (before measurement) probability of class \( C_i \).

\[
p(\mathbf{x}) = \sum_j p(\mathbf{x}|C_j)P(C_j)
\]

This gives the Bayes’ (or MAP) classifier:

\[
P(C_i|\mathbf{x}) \overset{C_i}{\gtrless} P(C_j|\mathbf{x})
\]

\[
p(\mathbf{x}|C_i)P(C_i) \overset{C_i}{\gtrless} p(\mathbf{x}|C_j)P(C_j)
\]

So assign the pattern \( \mathbf{x} \) to the class with the maximum weighted p.d.f.
The decision boundary $\theta$ is where

$$p(x|C_i)P(C_i) = p(x|C_j)P(C_j)$$

In 1-D (i.e. one feature) there are usually 2 boundaries. In n-D space, the decision surface is usually complicated.

The actual surfaces depend on the p.d.f.’s. If we restrict ourselves to Gaussian distributions, we can compare MAP to the other classifiers we’ve seen.
Consider the univariate Gaussian:

\[ p(x|C_i) = \frac{1}{\sqrt{2\pi\sigma_i}} e^{-\frac{1}{2} \left( \frac{x-\mu_i}{\sigma_i} \right)^2} \quad i = 1, 2, \ldots, N \]

It is convenient to use the ‘log-likelihood’ form since the p.d.f.’s involve exponentials.

\[
\begin{align*}
\frac{p(x|C_i)P(C_i)}{p(x|C_j)P(C_j)} & \geq 1 \\
\log \left( \frac{p(x|C_i)}{p(x|C_j)} \right) \frac{C_i}{C_j} & \geq \log \left( \frac{P(C_j)}{P(C_i)} \right) \\
C_i & \geq C_j
\end{align*}
\]

For the normal distribution this becomes:

\[
\left( \frac{x - \mu_j}{\sigma_j} \right)^2 - \left( \frac{x - \mu_i}{\sigma_i} \right)^2 \geq \frac{C_i}{C_j} \geq 2\log \left( \frac{P(C_j)\sigma_i}{P(C_i)\sigma_j} \right)
\]

Notes:

\[
\left( \frac{x - \mu_j}{\sigma_j} \right)^2 - \left( \frac{x - \mu_i}{\sigma_i} \right)^2 \geq \frac{C_i}{C_j} \geq 2\log \left( \frac{P(C_j)\sigma_i}{P(C_i)\sigma_j} \right)
\]

Quadratic in \( x \)

Measures distances from the class means in standard deviation units

When \( P(C_i) > P(C_j) \), the RHS > 0 and so \( C_i \) is preferred

When \( \sigma_i > \sigma_j \), RHS > 0 and so \( C_i \) is preferred

So MAP is biased towards favouring:

- most likely class
- most compact class
To find the decision boundary, set both sides equal and solve for $x$:

$$
\left( \frac{1}{\sigma^2_j} - \frac{1}{\sigma^2_i} \right) x^2 - 2 \left( \frac{\mu_j}{\sigma^2_j} - \frac{\mu_i}{\sigma^2_i} \right) x + \frac{\mu_j^2}{\sigma^2_j} - \frac{\mu_i^2}{\sigma^2_i} - 2 \log \left( \frac{P(C_j)\sigma_i}{P(C_i)\sigma_j} \right) = 0
$$

If $\sigma_i = \sigma_j$, there is a single boundary:

$$
2 \left( \frac{\mu_i - \mu_j}{\sigma^2} \right) x + \frac{\mu_j^2 - \mu_i^2}{\sigma^2} - 2 \log \left( \frac{P(C_j)}{P(C_i)} \right) = 0
$$

Solving for $x$ gives the value of $\theta$

$$
\theta = \frac{\mu_i + \mu_j}{2} + \frac{\sigma^2}{\mu_i - \mu_j} \log \left( \frac{P(C_j)}{P(C_i)} \right)
$$

Assuming $\mu_j > \mu_i$, then for $P(C_j) > P(C_i)$ the second term is negative and

$$
\theta = \frac{\mu_i + \mu_j}{2} - \delta
$$

Comparing $P(C_j) = 2 P(C_i)$

Thus the MAP decision boundary is biased towards the least likely class.
In general, $\sigma_i \neq \sigma_j$ and the MAP decision boundary equation has two solutions.

\[ \theta_1, \theta_2 \text{ are functions of } \mu_i, \mu_j, \sigma_i, \sigma_j, P(C_i), \text{ and } P(C_j) \]

Compare the MAP classifier with the MICD classifier:

**MAP:**

\[
\left( \frac{x - \mu_j}{\sigma_j} \right)^2 - \left( \frac{x - \mu_i}{\sigma_i} \right)^2 \overset{C_i}{\gtrless} \frac{2\log \left( \frac{P(C_j)\sigma_i}{P(C_i)\sigma_j} \right)}{C_j}
\]

**MICD:**

\[
\left( \frac{x - \mu_j}{\sigma_j} \right)^2 - \left( \frac{x - \mu_i}{\sigma_i} \right)^2 \overset{C_i}{\gtrless} 0
\]

So the MAP classifier is biased towards the most likely class and the most compact class.
Now for the multidimensional normal case (n features):

\[ p(x|C_i) = \frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} e^{\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)} \quad i = 1, 2, \ldots, N \]

Take the log-likelihood form (as in the 1-D case) to get the MAP classifier:

\[ (x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) - (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \approx \frac{C_i}{C_j} 2\log \left( \frac{P(C_j)|\Sigma_j|^{1/2}}{P(C_i)|\Sigma_i|^{1/2}} \right) \]

The decision surface is a hyperquadratic, but shifted from the MICD surface by:

\[ 2\log \left( \frac{P(C_j)}{P(C_i)} \right) + \log \left( \frac{|\Sigma_i|}{|\Sigma_j|} \right) \]

Note:

If we don’t know the a priori probabilities (or choose to ignore them), then we can drop the term \( 2\log \left( \frac{P(C_j)}{P(C_i)} \right) \)

The result is called the Maximum Likelihood (ML) Classifier.
Summary

For normal distribution with maximum likelihood estimates of the mean and covariance matrix, the classifiers are:

Loss Function

Often decisions based on classification have associated losses. The losses are not necessarily constant for each decision.

For example, misclassifying a pattern as class 1 when it is really class 2 might be ‘worse’ than the reverse case. This can occur when one class is very rare, but costly if missed (such as credit card fraud).

\[
p(x|C_1)P(C_1) \frac{1}{2} \geq p(x|C_2)P(C_2)
\]

if \( P(C_1) \gg P(C_2) \) then the lowest error rate can be attained by always classifying as \( C_1 \)
One option is to improve the selection of features such that 
$p(x|C_2) >> p(x|C_1)$ for some values of $x$. Although this is ideal, especially if perfect separation can be attained, in practice this might not be feasible.

Another option is to assign a loss to misclassification.

Let \( \{\alpha_1 \ldots \alpha_n\} \) be the finite set of actions based on classifications.

Can think of these actions as assigning a pattern to a class, but doesn’t have to be the case. Could be something like ‘sound alarm’.

**Conditional Risk**

\[
R(\alpha_i|x) = \sum_{j=1}^{N_C} \lambda(\alpha_i|C_j) P(C_j|x)
\]

\(N_C\) is the number of classes

The loss associated with action \(i\) when the true state is \(C_j\)

To minimize overall risk, choose the action with the lowest risk for the pattern:

\[
R(\alpha_i|x) \frac{\alpha_j}{\alpha_i} \geq R(\alpha_j|x)
\]

**Shorthand notation:**

\[
\lambda_{ij} \equiv \lambda(\alpha_i|C_j) \equiv \text{cost of action } i \text{ given class } j
\]
Loss for Two Classes (i)

Assume: \( \lambda_{11} = \lambda_{22} = 0 \) (no loss for correct choice)
\[ \lambda_{12} = \lambda_{21} = 1 \] (same loss for incorrect choice)

\[ R(\alpha_1 | x) \gtrless R(\alpha_2 | x) \]
\[ \lambda_{11} P(C_1 | x) + \lambda_{12} P(C_2 | x) \gtrless \lambda_{21} P(C_1 | x) + \lambda_{22} P(C_2 | x) \]
\[ P(C_2 | x) \gtrless P(C_1 | x) \]

This is the MAP classifier.

Loss for Two Classes (ii)

Assume: \( \lambda_{11} = \lambda_{22} = 0 \) (no loss for correct choice)
\[ \lambda_{12} \neq \lambda_{21} \] (different losses for incorrect choices)

\[ \lambda_{11} P(C_1 | x) + \lambda_{12} P(C_2 | x) \gtrless \lambda_{21} P(C_1 | x) + \lambda_{22} P(C_2 | x) \]

Use Bayes' theorem:
\[ \lambda_{12} p(x | C_2) P(C_2) \gtrless \lambda_{21} p(x | C_1) P(C_1) \]

So if \( \lambda_{12} > \lambda_{21} \), thus there is greater chance of loss in choosing action 1 when real class is C2. Therefore we are more likely to choose action 2.
**Example**

**Credit card fraud** is a big problem. A fraudster will obtain the credit card information of a victim and proceed to use the card to purchase as many items as possible (usually items that are easy to sell).

Credit card companies use real-time analysis of transactions to attempt to separate fraudulent activity from legitimate. The sooner fraud can be detected, the lower the cost to the credit card company, which has to assume almost all of the losses.

The analysis uses features from the transactions, including location, type, speed, and physical velocity. However, the features are not sufficient to perfectly classify a transaction as fraud or legitimate. There is always the risk of inconveniencing customers.

Assuming that the amount of fraudulent activity is about 1% of the total credit card activity:

- \( C_1 = \text{Fraud} \quad P(C_1) = 0.01 \)
- \( C_2 = \text{No fraud} \quad P(C_2) = 0.99 \)

If losses are equal for misclassification, then:

\[
p(x|C_1)^{\alpha_1} \geq 99 p(x|C_2)^{\alpha_2}
\]
Example

However, losses are probably not the same. Classifying a fraudulent transaction as legitimate leads to direct dollar losses as well as intangible losses (e.g. reputation, hassles for consumers).

Classifying a legitimate transaction as fraudulent inconveniences consumers, as their purchases are denied. This could lead to loss of future business.

Let’s assume that the ratio of loss for not fraud to fraud is 1 to 50.

\[
\lambda_{12} = 1 \quad \lambda_{21} = 50
\]

A missed fraud is 50 times more expensive than accidentally freezing a card due to legitimate use.
Example

\[ p(x|C_1) \overset{\alpha_1}{\gtrsim} \frac{99}{\lambda_{21}} p(x|C_2) \]

\[ p(x|C_1) \overset{\alpha_1}{\gtrsim} \frac{1}{50} p(x|C_2) \]

\[ p(x|C_1) \overset{\alpha_1}{\gtrsim} 2 p(x|C_2) \]

By including the loss function, the decision boundaries change significantly.

Figure 2.1 From Pattern Classification by Duda, Hart, and Stork

Example

Range is much wider.

Must be 2 times greater.